attached to IDS

backup copy submitted on 02/28/02 to complete the missing copy.

D.P. 03/5/02

Digital Signal Processing Applications with the TMS320 Family

Volume 1

Edited by Kun-Shan Lin, Ph.D.

Digital Signal Processing
Semiconductor Group
Texas Instruments

TEXAS INSTRUMENTS

INSIDE

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes in the devices or the device specifications identified in this publication without notice. TI advises its customers to obtain the latest version of device specifications to verify, before placing orders, that the information being relied upon by the customer is current.

In the absence of written agreement to the contrary, TI assumes no liability for TI applications assistance, customer's product design, or infringement of patents or copyrights of third parties by or arising from use of semiconductor devices described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor device might be or are used.

The software code contained in this book is copyrighted and all rights are reserved by Texas Instruments, Inc. This code is intended for use on a Texas Instruments digital signal processor (TMS32010, TMS32020 . . .). No other use is authorized.

Reprinted June 1989

TRADEMARKS

The trademarks that have been mentioned in this book are credited to the respective corporations in the listing below.

TRADEMARK	CORPORATION
Apple	Apple Computers, Inc.
CP/M	Digital Research, Inc.
CROSSTALK	Microstuf, Inc.
DEC, DECtalk	Digital Equipment Corp.
DFDP	Atlanta Signal Processors Inc.
Eclipse	Data General Corp.
EZ-PRO	American Automation
IBM	IBM Corporation
ILS	Signal Technology, Inc.
Intel	Intel Corporation
Microstuf	Microstuf, Inc.
MS-DOS	Microsoft, Inc.
PAL	Monolithic Memories, Inc.
PC-DOS	IBM Corporation
PDP-11,Q-Bus	Digital Equipment Corp.
TEKTRONIX	Tektronix Corporation
UNIX	Bell Laboratories
VAX,VMS	Digital Equipment Corp.
VMEBUS	Motorola, Inc.

Contents

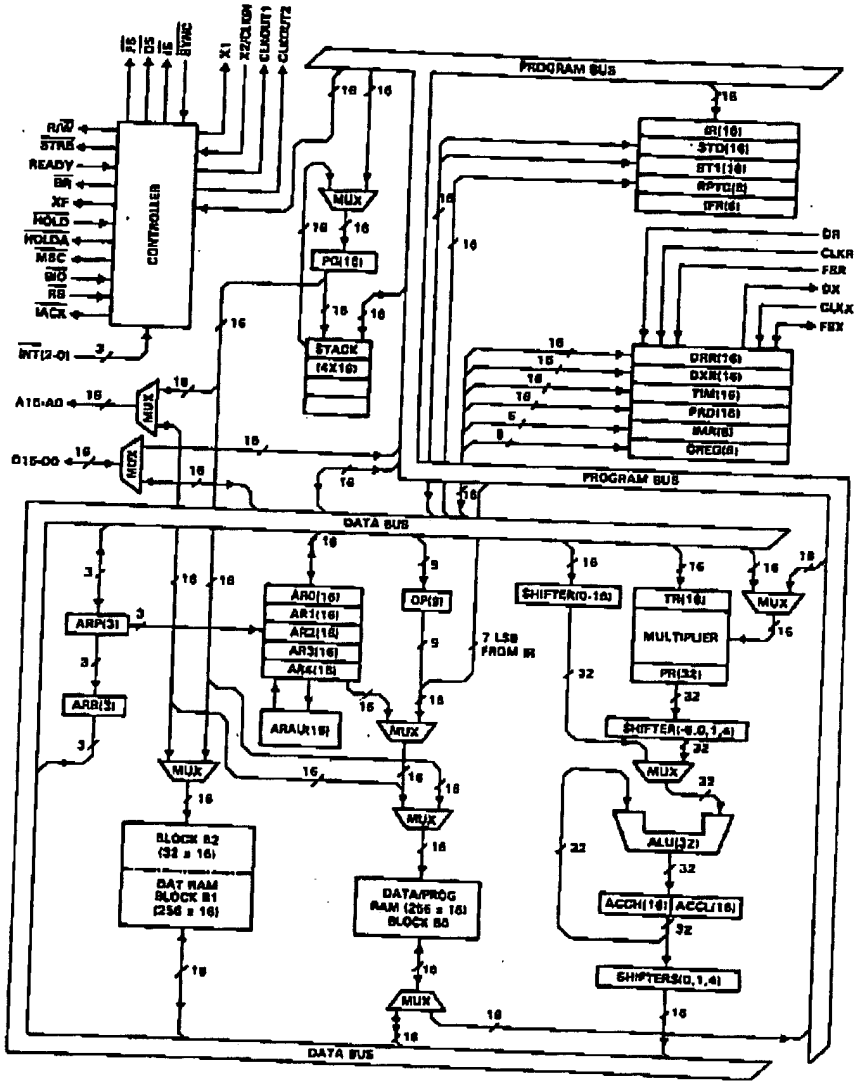
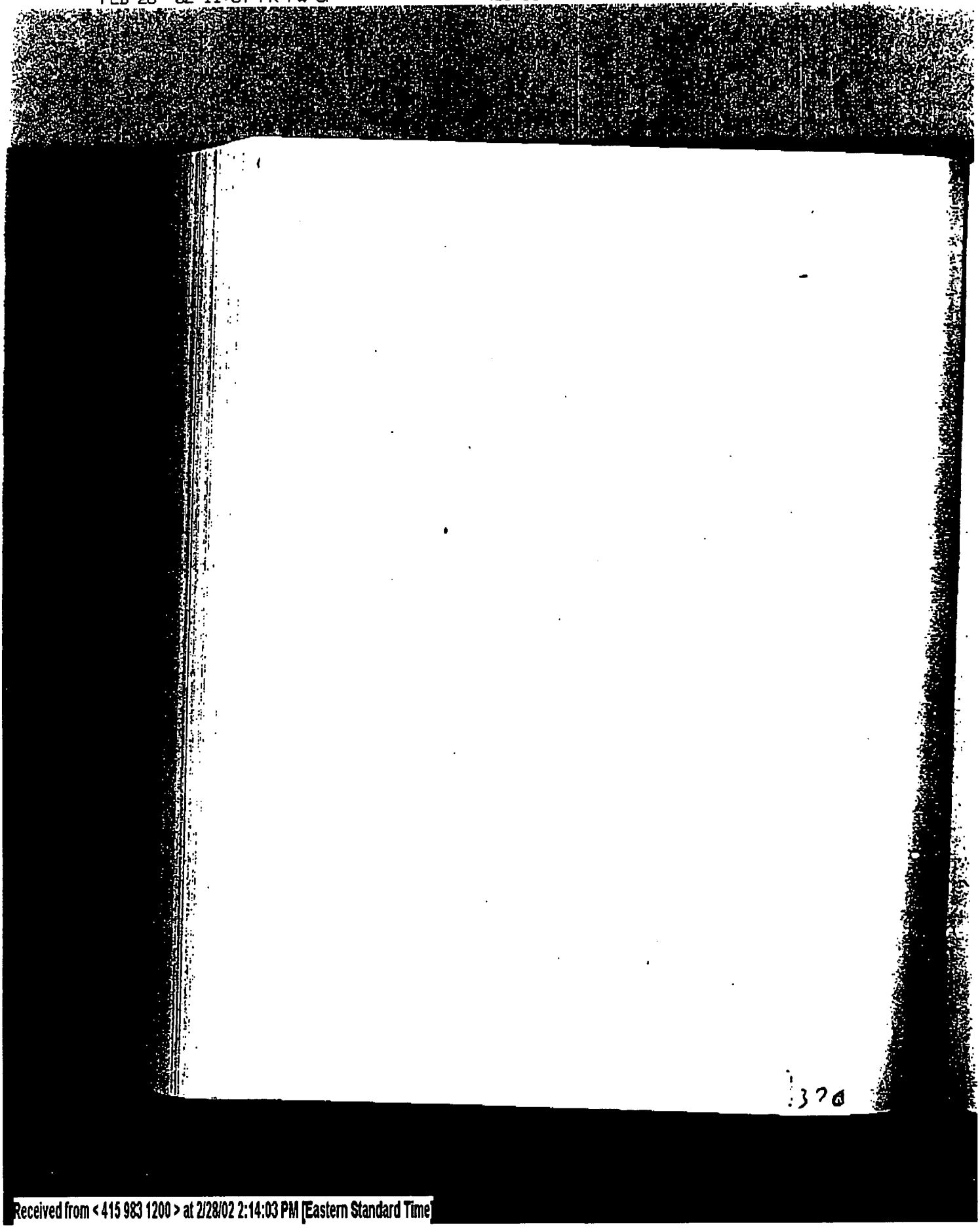


Figure 3. Functional Block Diagram of the TMS32020

13. TMS32020 and MC68000 Interface

Charles Crowell
Digital Signal Processing - Semiconductor Group
Texas Instruments



INTRODUCTION

Certain functions in a computer system may be too time consuming for a single processor to perform. A high-speed numeric processor, such as the TMS32020 Digital Signal Processor, may serve as a coprocessor with a slower yet capable host in a computer system. For example, many graphics algorithms must be implemented on a numeric coprocessor to the host so that the host can perform system functions while the coprocessor computes the numeric-intensive algorithms. The TMS32020 is capable of performing numeric functions, such as a multiply-accumulate, in a single cycle (200 ns). Other 16-bit processors, such as the Motorola MC68000, cannot approach the computational speed of the TMS32020, but have other qualities such as 'supervisor mode' and 'user mode' which make them useful as host processors.

This application report shows how the MC68000-10 can be used as a host processor with the TMS32020 serving as a numeric coprocessor to implement the numeric-intensive algorithms often required in computer systems. Applications for such a system include graphic workstations, speech processing, spectrum analysis, and other computational-intensive applications.

The schematic in the appendix has been fully built and tested and has proven functional.

SYSTEM CONFIGURATION

In Figure 1, the basic block diagram for the interface of the MC68000 with the TMS32020 is shown. The MC68000 is interfaced to its own separate program memory (EPROM) and data memory. Although the TMS32020 is interfaced to its own external program memory (PROM), it shares its external data memory with the MC68000. The TMS32020 typically has access to this shared data memory; however, the MC68000 can access this memory by asserting the $\overline{\text{HOLD}}$ line on the TMS32020. In this event, the TMS32020 places all its buses in a high-impedence state and turns on the buffers between the MC68000 and the shared data memory. This configuration allows the MC68000 to give the TMS32020 instructions and data, and then release the TMS32020 to perform various functions.

HARDWARE CONSIDERATIONS

Acknowledging Hold

After the MC68000 has written to the latch that puts the TMS32020 into the hold mode, the TMS32020 must communicate to the MC68000 that it is ready for the MC68000 to communicate with the shared data memory. The three methods of acknowledging hold to the MC68000 are as follows:

1. The MC68000 waits until it knows the TMS32020 is held.
2. The HOLD Acknowledge ($\overline{\text{HOLDA}}$) signal causes an interrupt to the MC68000.
3. The $\overline{\text{HOLDA}}$ signal writes to a memory-mapped latch.

The first method is implemented in the schematic in the appendix. This method assumes that the MC68000 will allow enough time for the buffers to be turned on before trying to access this memory. For example, the MC68000 could execute several NOP (No OPeration) instructions before attempting to access the shared data memory. Sometimes this method may not be sufficient. For example, if the TMS32020 is in the repeat mode, it does not recognize the $\overline{\text{HOLD}}$ assertion until it has finished the repeat instruction. This could cause a long unpredictable delay before the TMS32020 acknowledges the $\overline{\text{HOLD}}$ interrupt.

The second method of communicating to the MC68000 that the TMS32020 is in the hold mode is to interrupt the MC68000. To implement this, the $\overline{\text{HOLDA}}$ signal can be tied to one of the MC68000 interrupts, thus allowing the MC68000 to access the shared memory as fast as possible. Some method of communicating to the MC68000 as to which device caused the interrupt needs to be considered, since the MC68000 searches all external devices for the originator of the interrupt.

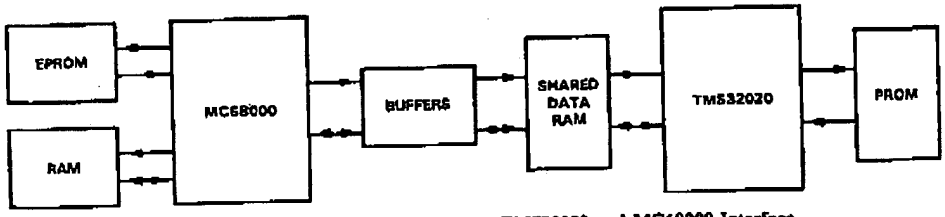


Figure 1. System Block Diagram of the TMS32020 and MC68000 Interface

The third method of communicating to the MC68000 when it can access the shared memory is to allow the $\overline{HOLDA}$ to be read through a memory-mapped latch. Then, the MC68000 can poll this memory location until it recognizes a change, thus signifying that the TMS32020 has indeed been placed in the hold mode.

Communicating with Shared Memory

Once the TMS32020 has acknowledged the $\overline{HOLD}$ assertion, the three-state buffers (74LS241) are turned on to allow the MC68000 address bus and R/W line to become valid to the shared memory (IMS1421-40). These buffers are physically enabled by $\overline{HOLDA}$, thus assuring that the TMS32020 has three-stated its memory bus. Once the address becomes valid, the transceivers (74LS245) are enabled so that the MC68000 data bus can access the shared memory. These buffers are enabled by the output of the decoder (74ALS138). By doing this, the MC68000 data bus accesses the shared-memory data bus only when MC68000 is trying to access the shared memory. This prevents data bus conflicts when the MC68000 accesses other memory while the TMS32020 is being held. After the communication path is enabled, the MC68000 can read and write to the shared memory. Figure 2 shows the timing when the MC68000 writes to the shared memory. The Enable/Select ($\overline{E/S}$) on the shared memory is enabled when the address and the Address Strobe ($\overline{AS}$) on the MC68000 become valid. The rising edge of the $\overline{AS}$ causes $\overline{E/S}$ to rise, thus writing to the shared memory.

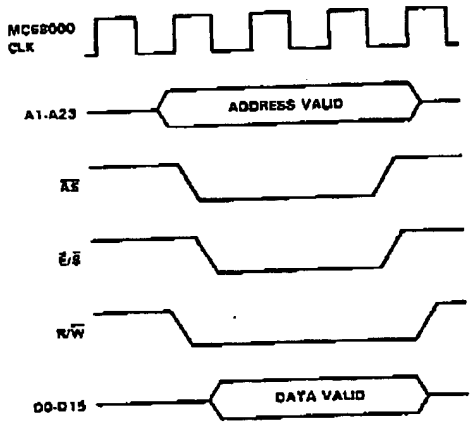


Figure 2. MC68000 Write Cycle to Shared Memory

When the TMS32020 is not in the $\overline{HOLD}$ mode, it can communicate directly with the shared data memory. The three-state buffers and the transceivers between the MC68000 and the shared data memory are turned off, and the link between the TMS32020 and the shared memory is direct. Figure 3 shows the timing when the TMS32020 writes to the shared memory. The $\overline{E/S}$ on the IMS1421-40 is enabled by Data Strobe ($\overline{DS}$) and Strobe ($\overline{STRB}$) becoming valid on the TMS32020. The rising edge of $\overline{STRB}$ causes $\overline{E/S}$ to rise, thus writing to the shared memory.

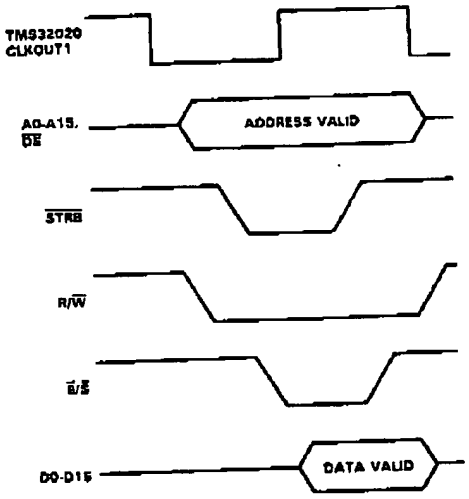


Figure 3. TMS32020 Write Cycle to Shared Memory

$\overline{LDS}$ and $\overline{UDS}$ Considerations

The schematic in the appendix is the expansion of the block diagram shown in Figure 1. In this schematic, the Upper Data Strobe ($\overline{UDS}$) and Lower Data Strobe ($\overline{LDS}$) signals on the MC68000 are not included, i.e., not connected. This method is sufficient if 'word'-specified instructions are the only ones used on the MC68000. Many systems work more efficiently if other length specifications for some of the MC68000 instructions are used. Therefore, a decode scheme, such as in Figure 4, may be implemented. In this scheme, the MC68000 can read or write bytes or words to the Synertek RAMs (SY2128). For example, the MC68000 may write to data bits D0-D7 and not affect the upper data bits by asserting $\overline{LDS}$ low and leaving $\overline{UDS}$ at a logic one.

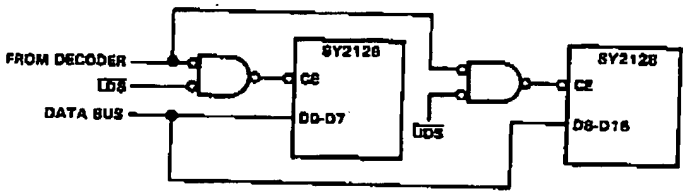


Figure 4. $\overline{LDS}$ and $\overline{UDS}$ Scheme for Memory Access

This is an automatic function of the MC68000 if 'byte' lengths are specified on certain instructions.

SUMMARY

The TMS32020 Digital Signal Processor is capable of performing numeric-intensive algorithms faster than other numeric coprocessors used in the past. In addition, the TMS32020 offers a minimal-chip, cost-effective solution to applications requiring a high-performance coprocessor.

This report shows how the TMS32020 can work with the MC68000 to serve as a numeric coprocessor. The interface shown in this report is a generic one and can be used with different host processors. A block diagram of the system configuration is included, as well as hardware considerations. The appendix contains a fully tested schematic of the design presented in this report.

APPENDIX

Schematic of TMS32020 and MC68000 Interface

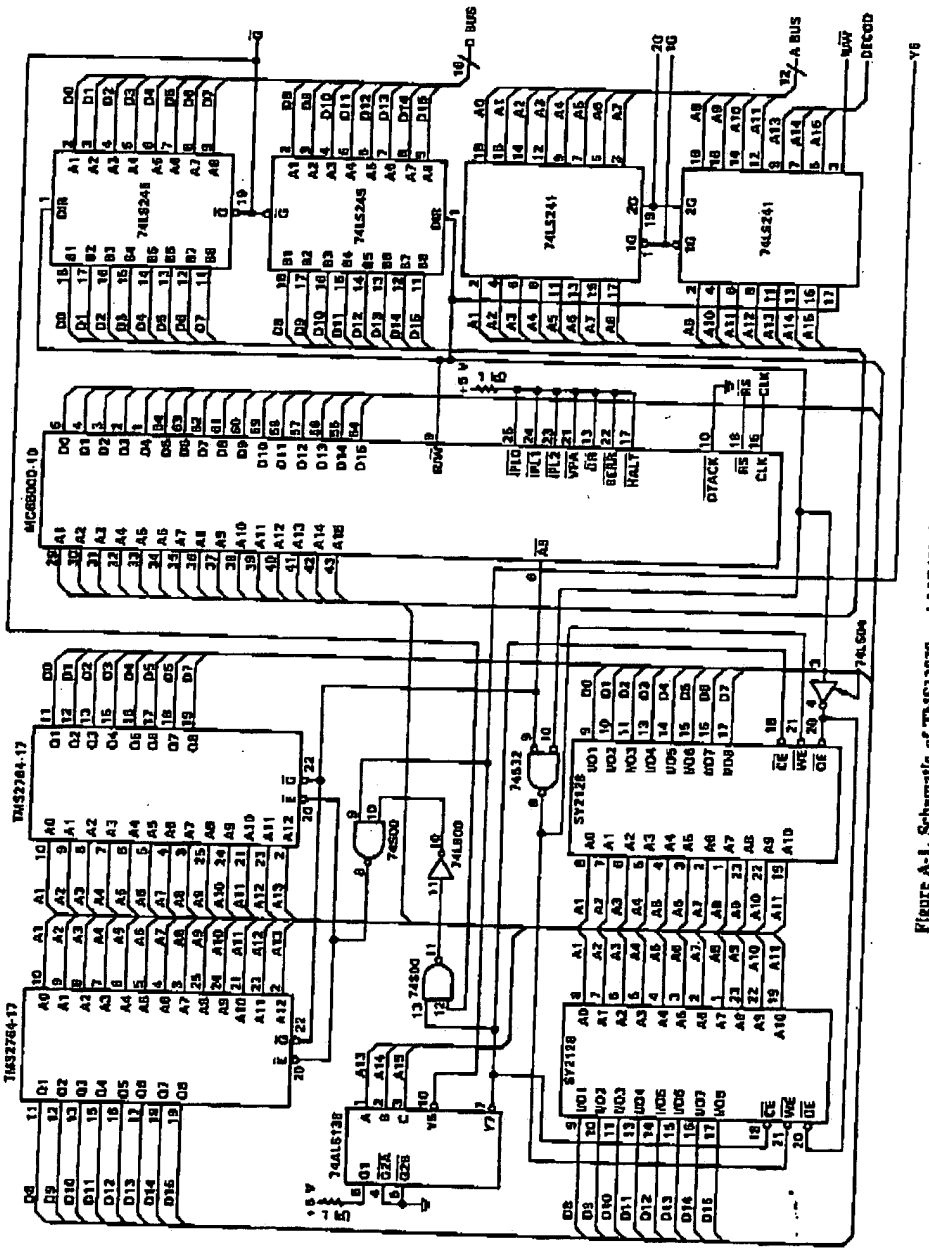


Figure A-1. Schematic of TMS32020 and MC68000 Interface (Sheet 1 of 3)

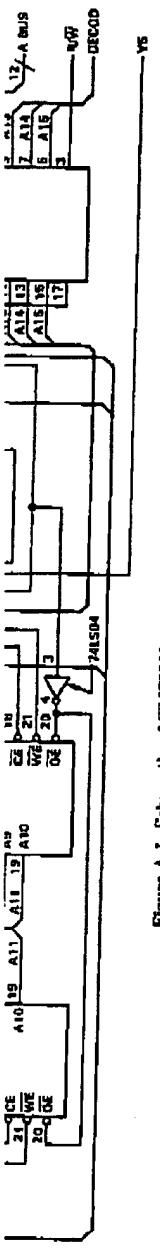


Figure A-1. Schematic of TMS32020 and MC68000 Interface (Sheet 1 of 3)

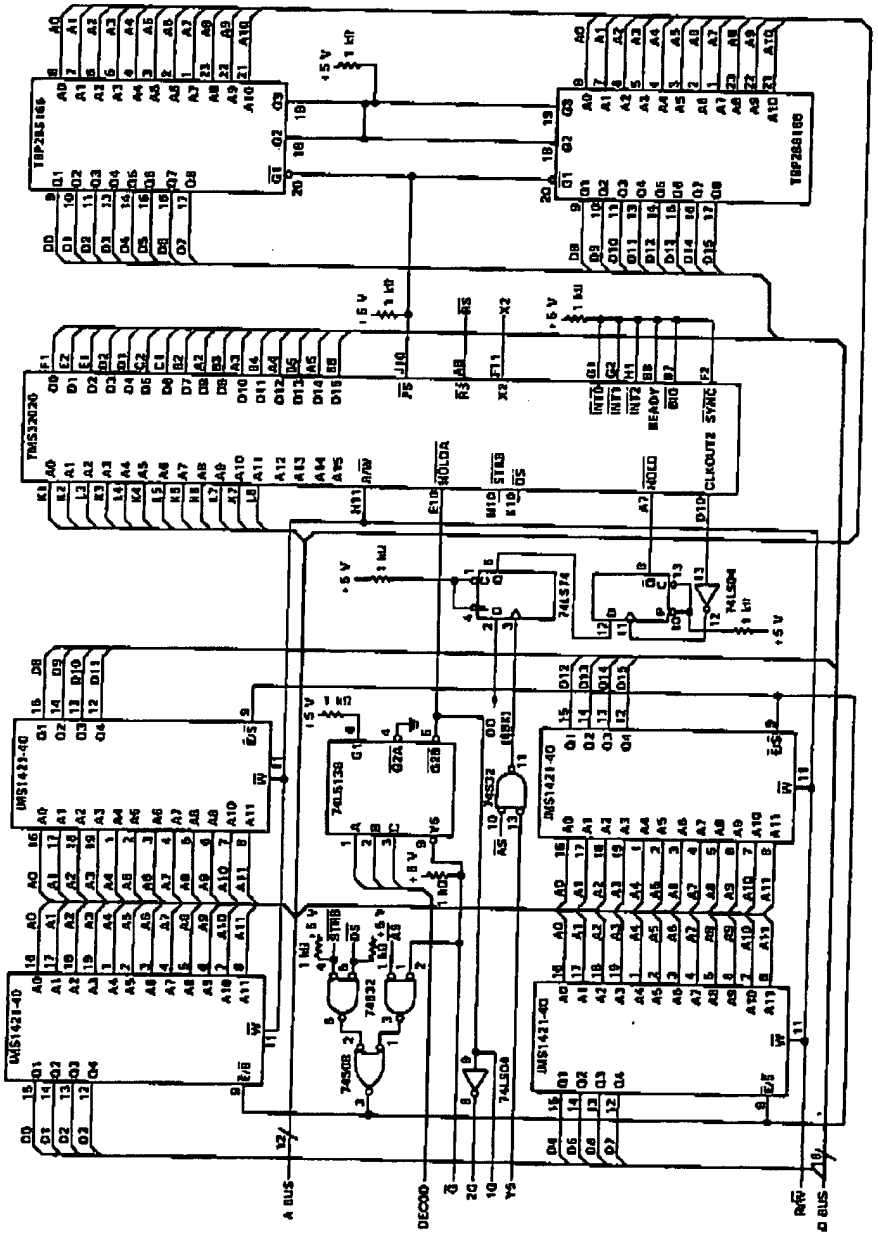


Figure A-1. Schematic of TMS32020 and MC68000 Interface (Sheet 2 of 3)

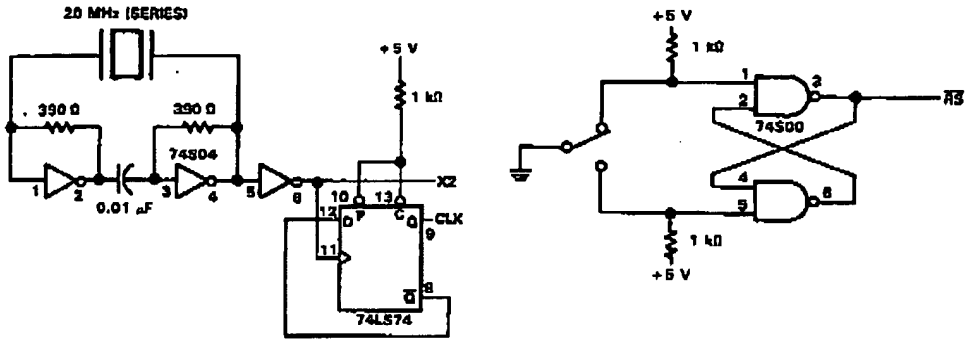


Figure A-1. Schematic of TMS32020 and MC68000 Interface (Sheet 3 of 3)

[redacted]

TEXAS INSTRUMENTS

Printed in U.S.A., September 1996

SPRA012A

BACK COVER